MXNet主要模块简介

NDArray

MXNet的NDArray提供了命令式的张量计算。相比numpy的ndarray，MXNet的NDArray独具两大特征。其一，对NDArray的操作可以在不同的设备上进行。MXNet的NDArray不仅存储了一个n维度数组，还包含一个context指向其存储的位置(cpu,gpu)。但要注意的是MXNet不支持对不同设备上的NDArray进行操作，即操作的NDArray的context必须一致。其二，在MXNet中对NDArray的操作都会被自动并行执行。这交由MXNet的引擎模块安排执行。

IO

IO模块实现了高效、分布式的数据加载和预处理功能。

在数据存储方面，IO模块采用了recordIO的格式，即每条数据都被存储为一个binary record。这样做的优点主要体现在以下三个方面：

1. 每条record的长度可以是不同的，因此数据的存储可以更加紧凑
2. 将数据打包之后，每个文件包含许多条数据，可以连续读取
3. 使用recordIO格式可以通用magic number定位到每条record的起始或结束位置，为数据分块提供了便利

另外IO模块使用了多线程来加速数据载入，用threadeditor隐藏IO开销:用一个线程来缓存数据，另一个线程从缓存区读取数据。缓存区数据被读取一部分，则自动缓存更多数据。

可以通过数据迭代器来获取数据。创建数据迭代器必须提供dataset参数和batch参数，其他参数可以根据需求进行设置。如果要在多个设备上进行训练，还需要创建kvstore，提供数据分块相关的参数。

KVStore

KVStore模块处理多设备、多机器训练时的参数同步问题。它的主要函数有push和pull。在每次push后将会对原本存储在store中的值进行更新。可以通过设置updater让数据以我们需要的方式进行更新。

在多机器分布式训练中，kvstore提供了同步更新和异步更新两种同步方式。

同步更新：

每个worker将算出的梯度push到store，kvstore对收到的数据进行求和，然后根据updater更新权重，然后每个worker通过pull获得更新后的权重。若有n个worker，每个worker的batch size为b，则同步更新的效果与在单一机器上batch size为n\*b的训练效果相当。

异步更新：

每个worker算出梯度后立刻更新。和同步更新相比，异步更新由于不用每次迭代都进行同步，因而速度更快，但每个worker更新时使用的是几个迭代之前的权重，因此收敛性不如同步更新。

为了保证数据一致性，kvstore中的操作也是由引擎控制执行的。

Runtime dependency engine

引擎根据传进操作的读写依赖安排执行这些操作。NDArray，symbol， kvstore的操作都是由引擎来安排执行的。

采用多线程，并行执行没有依赖冲突的操作。

Symbol construction

MXNet使用多值输出的符号表达式(symbol)来声明计算图。Symbol由变量和运算符构成，其中变量可以是自由变量，也可以是其他symbol的输出。MXNet将多种常用操作封装为operator，方便了网络的构建。另外MXNet还提供了计算图可视化、symbol的保存和载入等功能。

Symbolic execution

Symbolic execution模块主要完成计算图的优化和存储分配，并执行计算图的运算。

计算图优化：只保留计算图中对计算需要的输出有用的部分

存储分配：采用了inplace和co-share两种策略来压缩存储空间

为了执行计算图中的运算，必须给计算图中的自由变量绑定输入并声明需要的输出，这一操作可以通过bind或simple\_bind完成。

Inplace：首先为每个节点的输入和输出设置一个reference count，代表它在计算图中将被使用的次数，然后按拓扑顺序模拟遍历计算图，更新reference count的值。当一个输入的reference count变为1，则在本次操作之后不再需要该输入，如果此时还未给输出分配空间，则可以将输出保存在该输入所占的空间。对于reference count减少到0的输入和输出，释放它们的空间。这样通过模拟计算每个节点需要空间的大小，再执行实际内存分配。

co-share：允许不能并行执行的节点之间共享临时空间。采用贪婪法，每次找出计算图中最长的一条路径，将该路径上的节点标记为同一种颜色，这样相同颜色的节点相互之间存在依赖关系而无法同时进行，因此可以共享临时空间。

Bind操作后获得executor，便可通过forward函数计算输出，backward函数计算梯度。另外executor提供了回调函数用来监视每一步的输出(forward)或梯度(backward)用于调试。